

```

;;
;; ----- Creació nom capes -----
;;
(command "_LAYER" "_N" "Met1" "") ; Metall 1
(command "_LAYER" "_C" "120" "Met1" "")
(command "_LAYER" "_N" "Tox" "") ; Òxid de Porta + Primera capa de l'òxid de camp
(command "_LAYER" "_C" "50" "Tox" "")
(command "_LAYER" "_N" "Field" "") ; Òxid de camp
(command "_LAYER" "_C" "50" "Field" "")
(command "_LAYER" "_N" "Met2" "") ; Metall 2
(command "_LAYER" "_C" "170" "Met2" "")
(command "_LAYER" "_N" "Smc" "") ; Semiconductor Tipus "P"
(command "_LAYER" "_C" "210" "Smc" "")
(command "_LAYER" "_N" "Bb" "") ; Límit de la cel·la (Bounding box)
(command "_LAYER" "_C" "1" "Bb" "")

;;
;; ----- Normes de disseny. Totes les mides en nanometres
;;
(setq N201 60000) ; Ample mínim Semiconductor
(setq N501 160000) ; Ample mínim Metall 1
(setq N502 60000) ; Distància mínima entre Metalls 1
(setq N503 60000) ; Distància mínima Metall1 a Aïllant.
(setq N504 25000) ; Desbordament metall en contacte entre Metall1 i Metall2
(setq N601 160000) ; Ample mínim Metall 2
(setq N602 60000) ; Distància mínima entre Metalls 2
(setq N603 160000) ; Ample mínim finger (Drenador i Font)
(setq N604 240000) ; Desbordament mínim Metall1 sota Metall2 (fingers)
(setq N901 100000) ; Desbordament d'Aïllant sobre Metall 1 (intersecció Metall1 Metall2)
(setq N902 800000) ; Desbordament d'Aïllant sobre Metall 1 (fingers)

;;
;; ----- Paràmetres i constants
;;
;; Punt inserció (PARÀMETRES DE LA FUNCIÓ QUE INSERTARÀ LA CEL·LA)
(setq puntinsercio (list 0 0)) ; Punt d'inserció de la cel·la
(setq alcadacella 18860000) ; alçada cel·la
(setq Desbor 100000) ; desbordaments entrades i sortides per contactar fora de cel·la
(setq DesplXaS Desbor) ; veure figura.

;extrem esquerra inferior i superior
(setq ExtremEB puntinsercio)
(setq ExtremED (list (car ExtremEB) (+ (cadr ExtremEB) alcadacella)))

;;--inicialització dels límits de la cel·la (carrils d'alimentació) només coordinades en l'eix vertical

(setq amplada_carrils (* 4 N601)) ;; amplada dels carrils d'alimentació

;; Carril -Vss

(setq PmVssEB (cadr ExtremEB))
(setq PmVssDD (+ PmVssEB amplada_carrils))

;; Carril Gnd (extrem inferior de l'espai útil de la cel·la)
(setq pmGndEB (+ PmVssDD N503 N901))
(setq pmGndDD (+ PmGndEB amplada_carrils))

;; Carril Superior Vdd (extrem superior de l'espai útil de la cel·la)
(setq pmVddEB (- (cadr ExtremED) amplada_carrils))
(setq pmVddDD (cadr ExtremED))

;;
;; ----- Funcions
;;
;; Funcions d'accés a les coordenades dels punts (p1xy, p2xy)

(defun p1x (q)
  (car (car q)))

(defun p1y (q)

```

```
(cadr (car q)))

(defun p2x (q)
  (car (cadr q)))

(defun p2y (q)
  (cadr (cadr q)))
```

;; Funcions d'accés al contingut de l'estructura dotted list

```
(defun porta (tr)
  (cdr (assoc 'porta tr)))
(defun n_fingers (tr)
  (cdr (assoc 'nfingers tr)))
(defun aillant_tr (tr)
  (cdr (assoc 'aillant tr)))
```

;; Dibuixa els carrils d'alimentació

```
(defun carrils (ExtremDD / pveb pvdd)

  ;; - Vss
  (setq pveb ExtremEB)
  (setq pvdd (list (+ (car ExtremDD) Desbor) PmVssDD))
  (command "_RECTANGLE" pveb pvdd)
  (command "_ZOOM" "e")
  (command "_CHANGE" "_last" "" "_P" "_LA" "Met2" "")

  ;; - Gnd
  (setq pveb (list (car pveb) PmGndEB))
  (setq pvdd (list (car pvdd) PmGndDD))
  (command "_RECTANGLE" pveb pvdd)
  (command "_ZOOM" "e")
  (command "_CHANGE" "_last" "" "_P" "_LA" "Met2" "")

  ;; - Vdd
  (setq pveb (list (car pveb) PmVddEB))
  (setq pvdd (list (car pvdd) (cadr ExtremED)))
  (command "_RECTANGLE" pveb pvdd)
  (command "_ZOOM" "e")
  (command "_CHANGE" "_last" "" "_P" "_LA" "Met2" ""))

)
```

;;; -- contacte entrada-sortida [coordenada horitzontal]

```
(defun contacte_es (posX / PmEB PmDD)

  (setq PmEB (list posX (- (cadr ExtremEB) Desbor)))
  (setq PmDD (list (+ (car PmEB) N501) (+ (cadr ExtremEB) alcadacella Desbor)))
  (command "_RECTANGLE" PmEB PmDD)
  (command "_ZOOM" "e")
  (command "_CHANGE" "_last" "" "_P" "_LA" "Met1" ""))

  (list PmEB PmDD))
```

;;; - Contacte vertical metall 1  
 ;;; - pos = (extrem inferior (x,y), extrem superior)

```
(defun contacte1_V (pos amp / PmEB PmDD)

  (setq PmEB (list (car (car pos)) (cadr (car pos))))
  (setq PmDD (list (+ (car PmEB) N501 amp) (cadr pos)))

  (command "_RECTANGLE" PmEB PmDD)
  (command "_ZOOM" "e")
  (command "_CHANGE" "_last" "" "_P" "_LA" "Met1" ""))

  (list PmEB PmDD))
```

```
(defun contacte1_h (p1 p2 / PmEB PmDD)
```

```
(if (= (cadr p1) nil)
```

```

(setq PmEB (list (car p1) (- (cadr p2) N501)))
(setq PmEB p1)

(if (= (cadr p2) nil)
  (setq PmDD (list (car p2) (+ (cadr p1) N501)))
  (setq PmDD p2))

(command "_RECTANGLE" PmEB PmDD)
(command "_ZOOM" "e")
(command "_CHANGE" "_last" "" "_P" "_LA" "Met1" "")

(list PmEB PmDD))

(defun contacte2_h (p1 p2 / PmEB PmDD)

(if (= (cadr p1) nil)
  (setq PmEB (list (car p1) (- (cadr p2) N501)))
  (setq PmEB p1))

(if (= (cadr p2) nil)
  (setq PmDD (list (car p2) (+ (cadr p1) N501)))
  (setq PmDD p2))

(command "_RECTANGLE" PmEB PmDD)
(command "_ZOOM" "e")
(command "_CHANGE" "_last" "" "_P" "_LA" "Met2" "")

(list PmEB PmDD))

(defun contacte1_H2 (pos amp / PmEB PmDD)

  (setq PmEB (list (car (car pos)) (cadr (car pos))))
  (setq PmDD (list (car PmEB) (cadr pos)))

  (command "_RECTANGLE" PmEB PmDD)
  (command "_ZOOM" "e")
  (command "_CHANGE" "_last" "" "_P" "_LA" "Met1" "")

  (list PmEB PmDD))

(defun fingers_h (trs pos inv / tr num_fingers tmp1 tmp2 pfx1 pfx2 pfx3 pfx4 Px1 Px2 Py1 Py2 Pm2FEB Pm2FDD)

(setq num_fingers (n_fingers trs))
(setq tr (porta trs))

(setq tmp1 (p2x tr))
(setq tmp2 (p1x tr))

(if (= inv 1)
  (progn
    (setq pfx1 (cadr pos))
    (setq pfx2 (+ tmp1 N604))
    (setq pfx4 (car pos))
    (setq pfx3 (- tmp2 N604)))

  (progn
    (setq pfx1 (car pos))
    (setq pfx2 (- tmp2 N604))
    (setq pfx4 (cadr pos))
    (setq pfx3 (+ tmp1 N604))))

(setq Py1 (+ (p1y tr) N604))
(setq Px1 pfx3)
(setq Px2 pfx1)

(repeat num_fingers

  (setq Py2 (+ Py1 N603))
  (setq Pm2FEB (list Px1 Py1))
  (setq Pm2FDD (list Px2 Py2))
  (command "_RECTANGLE" Pm2FEB Pm2FDD)
  (command "_CHANGE" "_last" "" "_P" "_LA" "Met2" ""))

```

```
(setq Py1 (cadr Pm2FDD))
(setq Py2 (+ Py1 N201))
(setq PsmFEB (list (p1x tr) Py1))
(setq PsmFDD (list (p2x tr) Py2))
(command "_RECTANGLE" PsmFEB PsmFDD)
(command "_CHANGE" "_last" "" "_P" "_LA" "Smc" "")
```

```
(if (= Px1 Pfx3)
  (progn
    (setq Px1 Pfx4)
    (setq Px2 Pfx2))
```

```
(progn
  (setq Px1 Pfx3)
  (setq Px2 Pfx1)))
```

```
(setq Py1 Py2))
```

```
(setq Py2 (+ Py1 N603))
(setq Pm2FEB (list Px1 Py1))
(setq Pm2FDD (list Px2 Py2))
(command "_RECTANGLE" Pm2FEB Pm2FDD)
(command "_CHANGE" "_last" "" "_P" "_LA" "Met2" "")
(command "_Zoom" "e")
```

```
(defun fingers_v (trs pos inv / tr tmp1 tmp2 num_fingers pfy1 pfy2 pfy3 pfy4 Px1 Px2 Py1 Py2 Pm2FEB Pm2FDD)
```

```
(setq num_fingers (n_fingers trs))
(setq tr (porta trs))
```

```
(setq tmp1 (p1y tr))
(setq tmp2 (p2y tr))
```

```
(if (= inv 1)
  (progn
    (setq pfy1 (cadr pos))
    (setq pfy2 (- tmp1 N604))
    (setq pfy4 (car pos))
    (setq pfy3 (+ tmp2 N604)))
```

```
(progn
  (setq pfy1 (car pos))
  (setq pfy2 (+ tmp2 N604))
  (setq pfy4 (cadr pos))
  (setq pfy3 (- tmp1 N604))))
```

```
(setq Px1 (+ (p1x tr) N604))
(setq Py1 Pfy3)
(setq Py2 Pfy1)
```

```
(repeat num_fingers
```

```
(setq Px2 (+ Px1 N603))
(setq Pm2FEB (list Px1 Py1))
(setq Pm2FDD (list Px2 Py2))
(command "_RECTANGLE" Pm2FEB Pm2FDD)
(command "_CHANGE" "_last" "" "_P" "_LA" "Met2" ""))
```

```
(setq Px1 (car Pm2FDD))
(setq Px2 (+ Px1 N201))
(setq PsmFEB (list Px1 (p1y tr)))
(setq PsmFDD (list Px2 (p2y tr)))
(command "_RECTANGLE" PsmFEB PsmFDD)
(command "_CHANGE" "_last" "" "_P" "_LA" "Smc" ""))
```

```
(if (= Py1 Pfy3)
  (progn
    (setq Py1 Pfy4)
    (setq Py2 Pfy2))
```

```
(progn
  (setq Py1 Pfy3))
```

```

    (setq Py2 Pfy1)))

(setq Px1 Px2))

(setq Px2 (+ Px1 N603))
(setq Pm2FEB (list Px1 Py1))
(setq Pm2FDD (list Px2 Py2))
(command "_RECTANGLE" Pm2FEB Pm2FDD)
(command "_CHANGE" "_last" "" "_P" "_LA" "Met2" "")
(command "_Zoom" "e"))

(defun fingers (trs pos inv)

  (if (= (cdr (assoc 'rot trs)) 1)
      (fingers_h trs pos inv)
      (fingers_v trs pos inv)))

(defun W_trans (n)
  (+ (* n (+ N603 N201)) N601 N604 N604))

(defun dielect_transistor2 (p1 p2)
  (command "_RECTANGLE" p1 p2)
  (command "_ZOOM" "e")
  (command "_CHANGE" "_last" "" "_P" "_LA" "Tox" "")
  (list p1 p2))

(defun d_porta (p1 W wtr / Px Py Px2 Py2 PmGEB PmGDD)

  (setq Px (car p1))
  (setq Py (cadr p1))
  (setq Px2 (+ Px wtr))
  (setq Py2 (+ Py W))
  (setq PmGEB (list Px Py))
  (setq PmGDD (list Px2 Py2))
  (command "_RECTANGLE" PmGEB PmGDD)
  (command "_ZOOM" "e")
  (command "_CHANGE" "_last" "" "_P" "_LA" "Met1" ""))

  (list PmGEB PmGDD))

;; - Funció per a dibuixar un transistor
;; - px: punt horitzontal inici, py: punt vertical inici, params: paràmetres del tr: Wp Wptotal
;; params -> (longitud total, amplada transistor (W), longitud font / drenador, longitud canal)

(defun transistor (px py params rot / pxeb pyeb pxdd pydd amplada nfingers wtr fng temp1 temp2 amp_fing
amp_canal)

  (setq nfingers (/ (car params) (cadr params))) ;; Número de fingers = Wptotal / Wp
  (setq amplada (W_trans nfingers))
  (setq wtr (cadr params))

  (if (eq rot 1)
      (progn
        (setq temp1 wtr)
        (setq wtr amplada)
        (setq amplada temp1)))

  (setq temp1 (+ (* N902 2) amplada))
  (setq temp2 (+ (* N902 2) wtr))

  (if (= (car px) "ee")
      (progn
        (setq pxeb (cadr px))
        (setq pxdd (+ (cadr px) temp1))))

```

```

(if (= (car px) "ed")
  (progn
    (setq pxeb (- (cadr px) temp1))
    (setq pxdd (cadr px))))
(if (= (car py) "es")
  (progn
    (setq pyeb (- (cadr py) temp2))
    (setq pydd (cadr py))))
(if (= (car py) "ei")
  (progn
    (setq pyeb (cadr py))
    (setq pydd (+ (cadr py) temp2))))

(setq aill (dielect_transistor2 (list pxeb pyeb) (list pxdd pydd)))
(setq porta_tr (d_porta (list (+ pxeb N902) (+ pyeb N902)) wtr amplada))

```

```

(list
  (cons 'porta porta_tr)
  (cons 'aillant aill)
  (cons 'nfingers nfingers)
  (cons 'rot rot)
  (cons 'fingers amp_fing)
  (cons 'canal amp_canal)
  ;;(cons 'fingers fng)
  (cons 'wp wtr))

```

;; funció per obtenir el valor màxim d'amplada d'un o una llista de transistors

```

(defun maxtrX (tr)
  ;; - en el cas que t sigui un sol transistor (num params 3)
  (if (= (length tr) 3)
    (p2x (cdr (assoc 'aillant tr1)))
    (apply 'max (mapcar '(lambda (x) (p2x (cdr (assoc 'aillant x)))) tr))
  )
)

```

```

;; porta màxima
(defun maxtrXP (tr)
  ;; - en el cas que t sigui un sol transistor (num params 3)
  (if (= (length tr) 3)
    (p2x (cdr (assoc 'porta tr1)))
    (apply 'max (mapcar '(lambda (x) (p2x (cdr (assoc 'porta x)))) tr))
  )
)

```

```

;; funció genèrica
(defun maxtrXG (param tr)
  ;; - en el cas que t sigui un sol transistor (num params 3)
  (if (= (length tr) 3)
    (p2x (cdr (assoc param tr1)))
    (apply 'max (mapcar '(lambda (x) (p2x (cdr (assoc param x)))) tr))
  )
)

```

```

(defun contacte_porta (tr pos inici / pnt pnt_inici pnt_final pnt_mig PmEB PmDD)

  (setq pnt (cdr (assoc 'porta tr)))

  (if (= pos "v")
    (progn
      (setq pnt_mig (+ (/ (cdr (assoc 'wp tr)) 2) (p1y pnt)))
      (if (> (p1x pnt) inici)
        (progn
          (setq pnt_inici inici)
          (setq pnt_final (p1x pnt)))
        (progn
          (setq pnt_inici (p2x pnt))

```

```

        (setq pnt_final inici)))

    (setq pmEB (list pnt_inici (- pnt_mig (/ N501 2))))
    (setq pmDD (list pnt_final (+ pnt_mig (/ N501 2))))))

(if (= pos "h")
    (progn
      (setq pnt_mig (+ (/ (- (p2x pnt) (p1x pnt)) 2)))
      (if (> (p1y pnt) inici)
          (progn
            (setq pnt_inici inici)
            (setq pnt_final (p1y pnt)))
          (progn
            (setq pnt_inici (p2y pnt))
            (setq pnt_final inici)))

      (setq pmEB (list (+ (p1x pnt) (- pnt_mig (/ N501 2))) pnt_inici))
      (setq pmDD (list (+ (car pmEB) N501) pnt_final))))

(command "_RECTANGLE" PmEB PmDD)
(command "_ZOOM" "e")
(command "_CHANGE" "_last" "" "_P" "_LA" "Met1" ""))

(list PmEB PmDD))

(defun isP (q)

  (if (listp q)
      (progn
        (if (= (length q) 2)
            (foreach n (append
                        (mapcar 'listp (car q))
                        (mapcar 'listp (cadr q))) (not n))
              nil))
      nil))

(defun aillant (m / pxeb pyeb pxdd pydd PaiEB PaiDD)

  (if (isP m)
      (progn
        (setq PaiEB (list (- (p1x m) N901) (- (p1y m) N901)))
        (setq PaiDD (list (+ (p2x m) N901) (+ (p2y m) N901))))

      (progn

        (if (isP (car m))
            (progn
              (setq pxeb (- (p1x (car m)) N901))
              (setq pxdd (+ (p2x (car m)) N901)))
            (progn
              (setq pxeb (- (p1x (car (car m))) N901))
              (setq pxdd (+ (p2x (cadr (car m))) N901))))

        (if (isP (cadr m))
            (progn
              (setq pyeb (- (p1y (cadr m)) N901))
              (setq pydd (+ (p2y (cadr m)) N901)))
            (progn
              (setq pyeb (- (p1y (car (cadr m))) N901))
              (setq pydd (+ (p2y (cadr (cadr m))) N901))))

        (setq PaiEB (list pxeb pyeb))
        (setq PaiDD (list pxdd pydd))))

  (command "_RECTANGLE" PaiEB PaiDD)
  (command "_ZOOM" "e")
  (command "_CHANGE" "_last" "" "_P" "_LA" "Field" ""))

(list PaiEB PaiDD))

```

```

;; - comprova si c és una llista de polígons o només un
(defun isLP (c)

```

```

(if (listp c)
  (listp (car (car c)))
  nil))

(defun aillant_io (contacte / Pai1EB Pai1DD Pai2EB Pai2DD)

  (if (isLP contacte)
      (progn
        (setq Pai1EB (list (- (p1x (car contacte)) N901) (- PmVddEB N901)))
        (setq pai1DD (list (+ (p2x (cadr contacte)) N901) (+ PmVddDD N901)))

        (setq Pai2EB (list (- (p1x (car contacte)) N901) (- PmVssEB N901)))
        (setq Pai2DD (list (+ (p2x (cadr contacte)) N901) (+ PmGndDD N901))))

      (progn
        (setq Pai1EB (list (- (p1x contacte) N901) (- PmVddEB N901)))
        (setq pai1DD (list (+ (p2x contacte) N901) (+ PmVddDD N901)))

        (setq Pai2EB (list (- (p1x contacte) N901) (- PmVssEB N901)))
        (setq Pai2DD (list (+ (p2x contacte) N901) (+ PmGndDD N901))))))

  (command "_RECTANGLE" Pai1EB Pai1DD)
  (command "_ZOOM" "e")
  (command "_CHANGE" "_last" "" "_P" "_LA" "Field" "")

  (command "_RECTANGLE" Pai2EB Pai2DD)
  (command "_ZOOM" "e")
  (command "_CHANGE" "_last" "" "_P" "_LA" "Field" "")

  (list (list Pai1EB pai1DD) (list Pai2EB Pai2DD)))

(defun aillant_cont (tr contacte / PaiEB PaiDD)

  (setq PaiEB (list (- (p1x contacte) N901) (- PmGndEB N901)))
  (setq PaiDD (list (+ (p2x contacte) N901) (p1y (aillant_tr tr))))

  (command "_RECTANGLE" PaiEB PaiDD)
  (command "_ZOOM" "e")
  (command "_CHANGE" "_last" "" "_P" "_LA" "Field" "")

  (list PaiEB PaiDD))

(defun aillant_intersec (ch cv / p1 p2 PaiEB PaiDD)

  (setq p1 (inters (car ch) (list (p2x ch) (p1y ch))
                  (car cv) (list (p1x cv) (p2y cv))
                  )
        )
  )

  (setq p2 (inters (cadr cv) (list (p2x cv) (p1y cv))
                  (cadr ch) (list (p1x ch) (p2y ch))
                  )
        )
  )

  (setq PaiEB (list (- (car p1) N901) (- (cadr p1) N901)))
  (setq PaiDD (list (+ (car p2) N901) (+ (cadr p2) N901)))

  (command "_RECTANGLE" PaiEB PaiDD)
  (command "_ZOOM" "e")
  (command "_CHANGE" "_last" "" "_P" "_LA" "Field" "")

  (list PaiEB PaiDD)

  )

(defun punt_mig (d b)
  (/ (- d b) 2))

(defun contacte2_m (b d p1 p2 / PmEB PmDD)

  (setq PmEB (list p1 (+ (p2y b) (- (punt_mig (p1y d) (p2y b)) (/ N601 2))))))

```



```
(setq PmDD (list p2 (+ (cadr PmEB) N601)))
```

```
(command "_RECTANGLE" PmEB PmDD)
```

```
(command "_ZOOM" "e")
```

```
(command "_CHANGE" "_last" "" "_P" "_LA" "Met2" "")
```

```
(list PmEB PmDD))
```